# Problem 1

On the first line, you will receivea sequence of **pizza** **orders**. **Each order contains a different number of pizzas**, separated by comma and space **", "**. On the **second line**, you will receive a sequence of **employees** **with pizza-making capacities** (how much pizzas an employee could make), separated by comma and space **", "**.

Your task is to check if **all pizza orders are completed**.

To do that, you should take the **first order** and the **last** **employee** and see:

* If the number of pizzas in the order is **less than or equal to** the employee's pizza making capacity, the order is completed. **Remove** **both** the order and the employee.
* If the number of pizzas in the order is **greater than** the employee's pizza making capacity, the **remaining pizzas** from the order are going to be made by **the next employees** until the order is **completed**.
  + If there are **no more employees** to finish the order, consider it **not completed**.
* The restaurant **does not take** orders for more than **10 pizzas** **at once**.
* If an order is **invalid** (less than or equal to 0), you need to **remove it** **before** it is taken by an employee.

You should keep track of the **total pizzas that are being made**.

### Input

* On the **first line** you will be given a sequence of **pizza orders** each represented as a number – **integers** separated by comma and space **", "**
* On the **second line** you will be given a sequence of **employees** with pizza-making capacities – **integers** separated by comma and space **", "**

### Output

* If all orders are **successfully** completed, print:  
  **All orders are successfully completed!**

**Total pizzas made: {total count}**

**Employees: {left employees joined by ", "}**

* Otherwise, if you **ran out** of **employees** and there are still some **orders left** print:  
  **Not all orders are completed.  
  Orders left: {left orders joined by ", "}**

### Constraints

* You will always have **at least one order** and **at least one employee**
* All integers will be in range **[-100, 100]**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 11, 6, 8, 1  3, 1, 9, 10, 5, 9, 1 | All orders are successfully completed!  Total pizzas made: 15  Employees: 3, 1 |
| **Comment** | |
| 1) The restaurant do not take the first order for 11 pizzas.  2) The first employee (1) takes an order for 6 pizzas but could only make 1. 5 pizzas left.  3) The next employee (9) continues the same order for 5 pizzas. The order is completed. Remove both.  4) The next employee (5) takes an order for 8 pizzas but could only make 5. 3 pizzas left.  5) The next employee (10) continues the same order for 3 pizzas. The order is completed. Remove both.  6) The next employee (9) takes an order for 1 pizza. The order is completed. Remove both.  7) All orders are completed. | |
| **Input** | **Output** |
| 10, 9, 8, 7, 5  5, 10, 9, 8, 7 | Not all orders are completed.  Orders left: 2, 5 |
| **Comment** | |
| 1) The last employee (7) takes an order for 10 pizzas but could only make 7. 3 pizzas left.  2) The next employee (8) continues the same order for 3 pizzas. The order is completed. Remove both.  3) The next employee (9) takes an order for 9 pizzas. The order is completed. Remove both.  4) The next employee (10) takes an order for 8 pizzas. The order is completed. Remove both.  5) The next employee (5) takes an order for 7 pizzas but could only make 5. 2 pizzas left.  6) Orders are not completed. | |
| **Input** | **Output** |
| 12, -3, 14, 3, 2, 0  10, 15, 4, 6, 3, 1, 22, 1 | All orders are successfully completed!  Total pizzas made: 5  Employees: 10, 15, 4, 6 |